

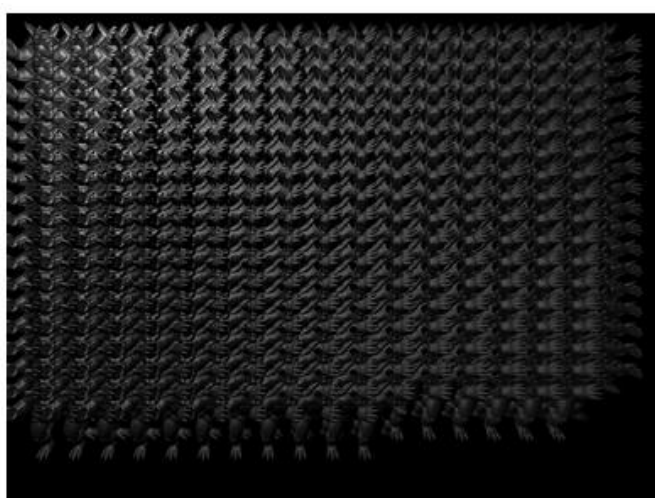
# Weekly report (2012.4.16~22)

## 上周回顾

到上周为止，已经基本实现了论文《Parallel Visualization on Large Clusters using MapReduce》中的基于 Map-Reduce 的绘制方法，但是绘制的规模上与原文还有较大的差距，绘制的最大场景规模只有 345,944 个面片。

## 本周进展

本周首先是在场景的规模上增大，方法是对原有的小规模场景进行复制，一共复制了 300 个 Armadillo 模型，使用 3600 个 MapTask 和一个 ReduceTask 进行绘制，以下是绘制结果(左边为结果图，右边为相关数据)，其中 max 2:14 表示的是 3600 个 MapTask 中耗时最多的使用了 2 分 14 秒。



Input	103.8M triangles
Output	20480*25360 (0.3G)
Map Task	3600
Reduce Task	1
Map Time	0:32:02 (max 2:14)
Reduce Time	2:32:53
Total	3:05:01

Figure 1 300Armadillo 绘制结果

分析结果可以明显地看到 ReduceTask 是一个瓶颈，所以之后对结果图采用分片处理的方式，分层 100 片，即使用 100 个 Reduce Task，结果如下，用时缩短了到 2:35:42。

Input	103.8M triangles
Output	20480*25360 (0.3G)
Map Task	3600
Reduce Task	1
Map Time	0:32:02 (max 2:14)
Reduce Time	2:32:53
Total	3:05:01
WorkLoad	25h

Input	103.8M triangles
Output	20480*25360 (0.3G)
Map Task	3600
Reduce Task	100
Map Time	1:34:27 (max 5:19)
Reduce Time	1:01:07 (max 54:31)
Total	2:35:42
WorkLoad	100h

Figure 2 (左) 分片前 (右) 分片后

再然后又对数据读取策略进行了改进，原先将场景下载到本地读取，但其实只需要其中的一小部分，现在改成从 DFS 直接读取所需要的部分。效率有所改善，但仍需 1 个多小时，

跑得最快的一次用时 1:17:32。

## 下周计划

绘制效率与原文有较大差距，一方面由于所使用的节点差异（原文中有 60 个节点，我只有 3 个节点），另一方面各种参数搭配可能也不够合理。下周将花一些时间调整参数，再看看能不能找到大点的集群。